

## Research Article

# Traffic Intensity Estimation in Finite Markovian Queueing Systems

Frederico R. B. Cruz <sup>1</sup>, Márcio A. C. Almeida <sup>2</sup>,  
Marcos F. S. V. D'Angelo <sup>3</sup>, and Tom van Woensel <sup>4</sup>

<sup>1</sup>Departamento de Estatística, Universidade Federal de Minas Gerais, 31270-901 Belo Horizonte, MG, Brazil

<sup>2</sup>Pró-Reitoria de Planejamento e Desenvolvimento, Universidade Federal do Pará, 66075-110 Belém, PA, Brazil

<sup>3</sup>Departamento de Ciência da Computação, Universidade Estadual de Montes Claros, 39401-089 Montes Claros, MG, Brazil

<sup>4</sup>Department of Industrial Engineering & Innovation Sciences, Eindhoven University of Technology, 5600 MB Eindhoven, Netherlands

Correspondence should be addressed to Frederico R. B. Cruz; [fcruz@est.ufmg.br](mailto:fcruz@est.ufmg.br)

Received 5 January 2018; Revised 10 April 2018; Accepted 16 May 2018; Published 26 June 2018

Academic Editor: Jason Gu

Copyright © 2018 Frederico R. B. Cruz et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In many everyday situations in which a queue is formed, queueing models may play a key role. By using such models, which are idealizations of reality, accurate performance measures can be determined, such as traffic intensity ( $\rho$ ), which is defined as the ratio between the arrival rate and the service rate. An intermediate step in the process includes the statistical estimation of the parameters of the proper model. In this study, we are interested in investigating the finite-sample behavior of some well-known methods for the estimation of  $\rho$  for single-server finite Markovian queues or, in Kendall notation,  $M/M/1/K$  queues, namely, the maximum likelihood estimator, Bayesian methods, and bootstrap corrections. We performed extensive simulations to verify the quality of the estimators for samples up to 200. The computational results show that accurate estimates in terms of the lowest mean squared errors can be obtained for a broad range of values in the parametric space by using the Jeffreys' prior. A numerical example is analyzed in detail, the limitations of the results are discussed, and notable topics to be further developed in this research area are presented.

## 1. Introduction

Queueing models are idealizations of many real systems. However, they enable the accurate determination of performance measures as long as a previous step has been fulfilled; that step is the statistical estimation of its parameters [1, 2]. It is impossible to discuss inference in queues without mentioning the pioneering work of Clarke in the 1950s, who describes maximum likelihood estimators for the arrival rates and service times in simple queues, and the work of Schruben and Kulkarni in the 1980s, who consider the problem of bias in queue estimation. It is also important to mention the Bayesian papers on the subject, including that of Muddapur in the 1970s, who published one of the first results that extended Clark's methodology, the series of papers from Armero and Bayarri [3, 4], Armero and Conesa [5–8], Choudhury and Borthakur [9], and, more recently,

Choudhury and Mukherjee [10, 11], Cruz et al. [12], and Quinino and Cruz [13]. These are only a few examples of the papers in this important research area.

The purpose of this paper is to evaluate the behavior of a traffic intensity estimator ( $\rho$ ), which is defined as the ratio between the arrival rate ( $\lambda$ ) and the service rate ( $\mu$ ), specifically for single-server finite Markovian queues, which model various real systems [14]. The type of queue researched in this paper is typically seen in many service-oriented settings, where there is a finite queue in front of a server. Think of a gas station, where cars can queue up for a limited amount of space, and the traffic intensity should be not too high, as in this case cars might not join the queue. An alternative example is observed within a supermarket context where customers line up for the check out. Having a good understanding of the traffic intensity is crucial in these situations. Good estimations are needed to properly design

```

MLERoMMIK<-function (K,samp) {
  loglike.f<-function (rho, K, n, sumxi) {
    n*log (1-rho)+sumxi*log (rho)-n*log (1-rho^(K + 1))}
  Eps.MLE<-1e-06
  n<-length (samp)
  sumxi<-sum (samp)
  res<-optimize (loglike.f,c(Eps.MLE,1-Eps.MLE), K, n, sumxi, maximum=TRUE, tol=Eps.MLE)
  return (res$maximum) }

```

LISTING 1: MLE for  $\rho$ .

the system (e.g., to install the gas pump buffer sizes needed) or to properly manage the system (e.g., to set waiting spaces for the store clerks at check out).

In summary, previous results obtained for *infinite* Markovian queues are extended here for *finite* Markovian queues. To reach this goal this paper combines some techniques and approaches from the work of Almeida and Cruz [15] (i.e., Bayesian inference and Monte Carlo simulation for evaluation of estimators under finite samples) with other classical tools (e.g., Gibbs sampling and bootstrapping).

The remainder of the paper is organized as follows. Section 2 details the queue equations and estimators for  $\rho$ . The computational results are presented and discussed in Section 3, followed by Section 4, which concludes the text with some final remarks and topics for future research in the area.

## 2. Material and Methods

When you have Poisson arrivals, exponential service times, a single server, and limited waiting space, you have an  $M/M/1/K$  queue; in Kendall notation,  $K$  represents the number of customers simultaneously allowed in the queueing system. The probability of a number  $X$  of users of the system, for  $x = 0, 1, \dots, K$ , is given by [14]:

$$P(X = x) \equiv p_x = \begin{cases} \frac{\rho^x (1 - \rho)}{1 - \rho^{K+1}}, & \text{for } \rho \neq 1, \\ \frac{1}{K + 1}, & \text{for } \rho = 1, \end{cases} \quad (1)$$

where  $\rho$  is the traffic intensity. Estimating traffic intensity is important as it is a key design parameter in production network design, routing of products, and so on.

**2.1. Maximum Likelihood Estimator.** Maximum likelihood estimation for the truncated geometric model is known since Thomasson and Kapadia [16]. Consider the stationary probability distribution given by (1). Next, consider a random sample of size  $n$ ,  $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ , where  $x_i$  is the number of customers an outside observer finds in the system. In this case, a maximum of  $K$  customers are allowed in the

queueing system at once. Therefore, the likelihood function is

$$\begin{aligned} L(\rho, \mathbf{x}) &= \frac{\rho^{x_1} (1 - \rho)}{1 - \rho^{K+1}} \times \dots \times \frac{\rho^{x_n} (1 - \rho)}{1 - \rho^{K+1}} \\ &= \frac{\rho^y \times (1 - \rho)^n}{(1 - \rho^{K+1})^n}, \end{aligned} \quad (2)$$

where  $y = \sum_{i=1}^n x_i$ . Note that the likelihood is a function of traffic intensity  $\rho$  and sample  $\mathbf{x}$ , although only its size,  $n$ , and its sum,  $y$ , which is a sufficient statistic for  $\rho$ , are necessary.

Needless to say that, for the implementation of maximum likelihood estimator (MLE), any bounded optimization algorithm could be used. However for the sake of simplicity, the implementation used in this study was encoded in R [17] and can be seen in Listing 1. For convenience, the logarithm of the likelihood function was considered because it allows products to be turned into sums. Maximizing the log-likelihood is done numerically through an R internal function. However, tests (not shown) were conducted with the original likelihood function; they indicated that the results did not change significantly in terms of accuracy or computational effort.

A nice feature about the MLEs is their *invariance* to transformations [18] in such a way that if  $\hat{\theta}$  is the MLE of  $\theta$  and  $\alpha = h(\theta)$  is a function of  $\theta$ , then  $\hat{\alpha} = h(\hat{\theta})$  is the MLE of  $\alpha$ . Thus, the expected number of customers in the system and the average queue length have, respectively, the following MLEs [14]:

$$\hat{L} = \frac{\hat{\rho}}{1 - \hat{\rho}} - \frac{(K + 1) \hat{\rho}^{K+1}}{1 - \hat{\rho}^{K+1}}, \quad (3)$$

$$\hat{L}_q = \frac{\hat{\rho}}{1 - \hat{\rho}} - \hat{\rho} \frac{1 + K \hat{\rho}^K}{1 - \hat{\rho}^{K+1}}, \quad (4)$$

in which  $\hat{\rho}$  is the MLE for  $\rho$ .

**2.2. Bayesian Inference.** One of the alternatives for making inferences is the Bayesian method. One of its main differences from the classical method is that Bayesian inference allows the incorporation of some *a priori* information into the model of the unknown parameters. Unlike the classical method, the Bayesian method considers these parameters random variables, i.e., associates them with probability distributions.

```

require (HI)
EBaRoMMIK<-function (K, samp, a, b) {
  logpost.f<-function (rho, K, n, sumxi, a, b) {
    (sumxi+a - 1)*log (rho)+(n+b - 1)*log (1-rho)-n*log (1-rho^(K + 1))}
  sSize<-5000
  n<-length (samp)
  sumxi<-sum (samp)
  res<-arms(runif (1), logpost.f, function (x, K, n, sumxi, a, b)((x > 0)*(x < 1)), sSize, K, n,
    sumxi, a, b)
  return (mean(res))}

```

LISTING 2: Bayes estimator for  $\rho$  for beta prior.

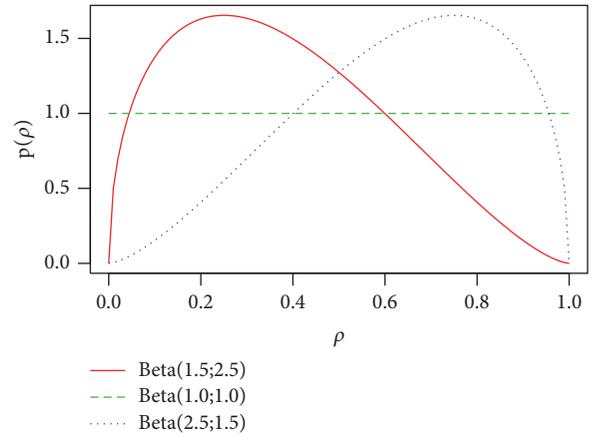
Therefore, the knowledge that the manager has regarding a given unknown parameter can be considered. Two different prior distributions are described as follows. However, other alternatives are possible, such as those proposed in the study by Armero and Bayarri [3]. Lingappaiah [19] also considered a Bayesian approach to this distribution.

**2.2.1. Beta Prior.** Therefore, the inference process for estimating  $\rho$  starts with (2) and assumes an *a priori* beta distribution, that is,  $p(\rho) \sim \text{Beta}(a, b)$ , which has been successfully used in inference in other Markovian queues [12, 13] and results in the following *a posteriori* distribution:

$$\begin{aligned}
 p_1(\rho | X) &\propto L(\rho, \mathbf{x}) \times p(\rho) \\
 &\propto \frac{\rho^y (1-\rho)^n}{(1-\rho^{K+1})^n} \\
 &\quad \times \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \rho^{(a-1)} (1-\rho)^{(b-1)}, \quad (5) \\
 &\propto \frac{\rho^{y+a-1} (1-\rho)^{n+b-1}}{(1-\rho^{K+1})^n}.
 \end{aligned}$$

Because the *a posteriori* distribution of  $\rho$  (5) is not a known distribution, it is necessary to use an approximation method to generate samples from the distribution. The implementation of the Bayesian estimator for beta prior is shown in Listing 2. Note that this is a bounded one-dimensional problem, and numerical integrations could be used as well to find the *a posteriori* distribution, which should be followed by another numerical integration to compute the estimator. An open research question is whether or not one method could be consistently superior to the other depending on the sample sizes in Gibbs sampling and the precision of the two numerical integrations.

A sample is extracted from the *a posteriori* distribution by using the function *arms* (for size 5,000), which is available in R's HI package [20]. The *a posteriori* distribution was represented by the logarithm of the probability density function (without the normalization constant). Because a quadratic loss function is considered, the Bayes point estimator is simply the average of the sample. Examples of *a priori* beta distributions are shown in Figure 1.

FIGURE 1: Beta prior distributions for traffic intensity  $\rho$ .

**2.2.2. Jeffreys Prior.** Someone might argue that it would be more natural to use a noninformative Jeffreys' prior distribution, which is defined in terms of the Fisher information given by

$$I(\rho) = E \left[ -\frac{\partial^2 \log p(X | \rho)}{\partial \rho^2} \right]. \quad (6)$$

Thus, the following prior distribution for  $\rho$  can be obtained:

$$p(\rho) \propto [I(\rho)]^{1/2}. \quad (7)$$

Thus, from (1), a  $p(\rho)$  can be found as follows. The logarithm of  $p(X | \rho)$ , from (1), is given by

$$\log p(X | \rho) = X \log \rho + \log(1-\rho) - \log(1-\rho^{K+1}), \quad (8)$$

for  $\rho \neq 1$ .

First and second derivatives are given, respectively, by

$$\frac{\partial \log p(X | \rho)}{\partial \rho} = \frac{X}{\rho} - \frac{1}{1-\rho} + \frac{(K+1)\rho^K}{1-\rho^{K+1}}, \quad (9)$$

```

require(HI)
EJeRoMMIK<-function (K, samp) {
  logJe.f<-function (rho, K, n, sumxi) {
    logpost<-(sumxi)*log (rho)+n*log(1-rho)-n*log(1-rho^(K+1))
    Irho<-(1/rho^2)*(rho/(1-rho)-(K+1)*rho^(K+1)/(1-rho^(K+1))) -
      1/(1-rho)^2 - (K+1)*(K*rho^(-K-1)+1)/(rho^(-K)-rho)^2
    if (!(is.nan(Irho)) && (Irho>0)) {
      logpost<-logpost+0.5*log(Irho) }
    return(logpost) }
  sSize<-1000
  n<-length(samp)
  sumxi<-sum(samp)
  res<-arms(runif(1), logJe.f, function(x, K, n, sumxi)((x > 0)*(x < 1)), sSize, K, n, sumxi)
  return(mean(res)) }

```

LISTING 3: Bayes estimator for  $\rho$  for Jeffreys' prior.

$$\frac{\partial^2 \log p(X | \rho)}{\partial \rho^2} = -\frac{X}{\rho^2} + \frac{1}{(1-\rho)^2} + \frac{(K+1)(K\rho^{-K-1} + 1)}{(\rho^{-K} - \rho)^2} \times \left[ \frac{1}{\rho^2} \left( \frac{\rho}{1-\rho} - \frac{(K+1)\rho^{K+1}}{1-\rho^{K+1}} \right) - \frac{1}{(1-\rho)^2} - \frac{(K+1)(K\rho^{-K-1} + 1)}{(\rho^{-K} - \rho)^2} \right]^{1/2}, \quad (10)$$

It follows that

$$\begin{aligned} I(\rho) &= E \left( -\frac{\partial^2 \log p(X | \rho)}{\partial \rho^2} \right) \\ &= E \left[ \frac{X}{\rho^2} - \frac{1}{(1-\rho)^2} - \frac{(K+1)(K\rho^{-K-1} + 1)}{(\rho^{-K} - \rho)^2} \right] \\ &= \frac{E(X)}{\rho^2} - \frac{1}{(1-\rho)^2} - \frac{(K+1)(K\rho^{-K-1} + 1)}{(\rho^{-K} - \rho)^2}. \end{aligned} \quad (11)$$

The expectation  $E[X]$  is given by [14]

$$E[X] = \frac{\rho}{1-\rho} - \frac{(K+1)\rho^{K+1}}{1-\rho^{K+1}}. \quad (12)$$

Then

$$p(\rho) \propto [I(\rho)]^{1/2} = \left[ \frac{1}{\rho^2} \left( \frac{\rho}{1-\rho} - \frac{(K+1)\rho^{K+1}}{1-\rho^{K+1}} \right) - \frac{1}{(1-\rho)^2} - \frac{(K+1)(K\rho^{-K-1} + 1)}{(\rho^{-K} - \rho)^2} \right]^{1/2}. \quad (13)$$

Thus, combining the likelihood, (2), and Jeffreys prior, given by (13), it is possible to find the following posterior probability distribution for  $\rho$ :

$$p_2(\rho | \mathbf{x}) \propto L(\rho, \mathbf{x}) \times p(\rho) \propto \frac{\rho^y \times (1-\rho)^n}{(1-\rho^{K+1})^n}$$

in which  $0 < \rho < 1$ .

Note that, unlike the case shown earlier, (5), the posterior distribution given by (14) is fixed. Indeed, it is not possible to vary Jeffreys' prior, which assumes a specific form. The implementation of the Bayesian estimator for Jeffreys' prior is shown in Listing 3. Because the logarithm of the posterior distribution is used, variable `Irho`, as defined in Listing 3, must be such that  $0 < \text{Irho} < \infty$ .

**2.3. Bootstrap Correction.** Among the bias correction methods that are commonly used for estimators, the bootstrap method is widely used [21]. In its nonparametric version, the method consists of making several (usually  $B = 1,000$ ) resamplings  $\mathbf{x}^*$  (with replacement). The parameter of interest is reestimated by some (usually biased) method,  $\widehat{\Theta}^*$ . Then, the average is calculated,  $\overline{\Theta}^*$ , and the bias is estimated using

$$\text{bias} = \overline{\Theta}^* - \widehat{\Theta}, \quad (15)$$

where  $\widehat{\Theta}$  is the estimate obtained from the original sample. Then, the following corrected version of the estimator is obtained:

$$\widetilde{\Theta}_B = 2\widehat{\Theta} - \overline{\Theta}^*. \quad (16)$$

The procedure is illustrated in Figure 2. The R code [17] for the bootstrap corrected estimator is shown in Listing 4. The parameter  $K$  is the maximum number of users simultaneously (in service and waiting) in the  $M/M/1/K$  queue. Note that 1,000 bootstrap replicates are used and that the correction occurs as part of the MLE (see Listing 1).

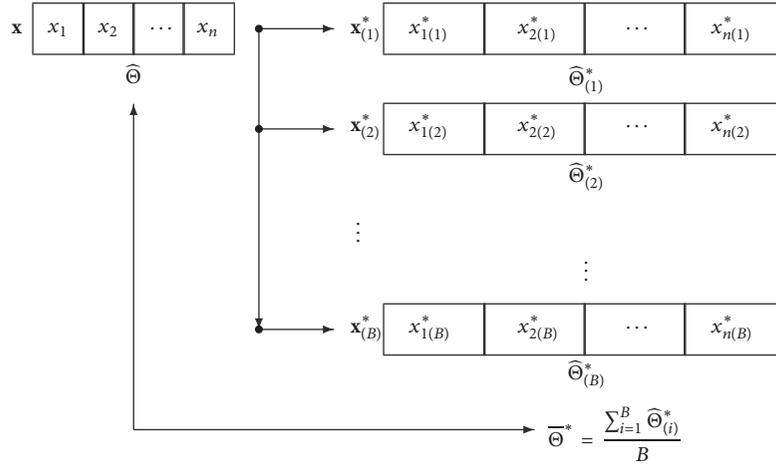


FIGURE 2: The bootstrap method.

```

EBoRoMMIK<-function(K, samp) {
  B<-1000
  summ<-0
  for (i in 1:B) {
    resamp<-sample(samp, replace=T)
    estm<-MLERoMMIK(K, resamp)
    summ<-summ+estm}
  estmStar=summ/B
  return(2*MLERoMMIK(K, samp)-estmStar) }
    
```

LISTING 4: Bootstrap corrected estimator.

Besides bias correction, the bootstrap method has been used by many researchers in the past with good results in confidence interval building and hypothesis testing [22]. As an example, an empirical bootstrap confidence interval is used in this work as a simple way of interval estimation for  $\rho$ . If the distribution of  $\delta = \hat{\Theta} - \Theta$  was known then the critical values  $\delta_{\alpha/2}$  and  $\delta_{1-\alpha/2}$  could be found, where  $\delta_\gamma$  is its  $\gamma$ 100%th percentile, and then

$$\Pr(\delta_{\alpha/2} \leq \hat{\Theta} - \Theta \leq \delta_{1-\alpha/2} \mid \Theta) = (1 - \alpha) \iff \quad (17)$$

$$\Pr(\hat{\Theta} - \delta_{\alpha/2} \geq \Theta \geq \hat{\Theta} - \delta_{1-\alpha/2} \mid \Theta) = (1 - \alpha),$$

which gives an  $(1 - \alpha)$ 100% confidence interval of

$$CI_{\Theta, (1-\alpha)100\%} = [\hat{\Theta} - \delta_{1-\alpha/2}, \hat{\Theta} - \delta_{\alpha/2}]. \quad (18)$$

The bootstrap makes it possible to estimate the distribution of  $\delta$  by the distribution of  $\delta^* = \hat{\Theta}^* - \hat{\Theta}$ , where  $\hat{\Theta}^*$  is the estimate obtained from an *empirical bootstrap sample* as explained earlier.

**2.4. Simulation  $M/M/1/K$  of Queues.** The number of users present in an  $M/M/1/K$  queue follows the distribution given by (1). To efficiently generate random variables from a discrete

distribution several methods are widely used in the literature, including function `sample`, from R [17]. The method used here is the discrete analog of the inverse transformation method, in which it is necessary to generate numbers  $R \sim \text{Unif}(0, 1)$ , *i.e.*, from a uniform distribution between 0 and 1, and to know the probabilities of interest,  $P\{X = j\} = p_j, \forall j$ . Therefore, to simulate a discrete random variable  $X$  with the probability function

$$P\{X = j\} = p_j, \quad j = 0, 1, \dots, \sum_{\forall j} p_j = 1, \quad (19)$$

it is necessary to compute

$$X = \begin{cases} 0, & \text{if } R \leq p_0, \\ 1, & \text{if } p_0 < R \leq p_0 + p_1, \\ \vdots & \\ j, & \text{if } \sum_{i=0}^{j-1} p_i < R \leq \sum_{i=0}^j p_i, \\ \vdots & \end{cases} \quad (20)$$

because

$$P\{X = j\} = P\left\{\sum_{i=0}^{j-1} p_i < R \leq \sum_{i=0}^j p_i\right\} = p_j, \quad (21)$$

and  $X$  follows the required probability distribution.

For  $M/M/1/K$  queues and from (1), the following must hold:

$$\sum_{i=0}^n p_i = \frac{1 - \rho^{n+1}}{1 - \rho^{K+1}}, \quad n = 0, 1, \dots, K. \quad (22)$$

Setting  $c = 1 - \rho^{K+1}$ , it follows that

$$\sum_{i=0}^{X-1} p_i < R \leq \sum_{i=0}^X p_i \implies$$

TABLE 1: Mean estimates for  $\rho$  and the mean squared error (MSE, in parentheses) for  $K = 5$ .

Estimate	$\rho$	$n$				
		10	20	50	100	200
MLE	0.01	0.00928 (0.00084)	0.00960 (0.00045)	0.01000 (0.00019)	0.01003 (0.00010)	0.01007 (0.00005)
	0.10	0.09172 (0.00715)	0.09614 (0.00384)	0.09845 (0.00160)	0.09907 (0.00081)	0.09964 (0.00041)
	0.20	0.18449 (0.01214)	0.19235 (0.00644)	0.19700 (0.00258)	0.19833 (0.00130)	0.19921 (0.00065)
	0.50	0.48492 (0.01946)	0.49422 (0.00932)	0.49824 (0.00358)	0.49884 (0.00180)	0.49952 (0.00088)
	0.90	0.87146 (0.01703)	0.88854 (0.00953)	0.89839 (0.00467)	0.90021 (0.00271)	0.90044 (0.00142)
	0.99	0.92503 (0.01488)	0.94638 (0.00734)	0.96396 (0.00303)	0.97225 (0.00159)	0.97854 (0.00083)
Beta	0.01	0.09043 (0.00699)	0.05378 (0.00226)	0.02864 (0.00052)	0.01953 (0.00018)	0.01487 (0.00007)
	0.10	0.15663 (0.00793)	0.13021 (0.00396)	0.11251 (0.00160)	0.10617 (0.00080)	0.10321 (0.00041)
	0.20	0.23444 (0.00989)	0.21739 (0.00567)	0.20690 (0.00244)	0.20327 (0.00126)	0.20167 (0.00064)
	0.50	0.50855 (0.01688)	0.50669 (0.00913)	0.50295 (0.00357)	0.50116 (0.00180)	0.50067 (0.00088)
	0.90	0.81770 (0.01385)	0.85318 (0.00654)	0.88214 (0.00283)	0.89387 (0.00178)	0.89954 (0.00114)
	0.99	0.85425 (0.02303)	0.89298 (0.01185)	0.92818 (0.00491)	0.94668 (0.00248)	0.96038 (0.00121)
Jeffreys	0,01	0,05000 (0,00215)	0,03182 (0,00084)	0,01931 (0,00026)	0,01476 (0,00012)	0,01244 (0,00005)
	0,10	0,11630 (0,00497)	0,10928 (0,00314)	0,10422 (0,00149)	0,10202 (0,00078)	0,10113 (0,00040)
	0,20	0,19457 (0,00908)	0,19512 (0,00520)	0,19804 (0,00229)	0,19907 (0,00123)	0,19963 (0,00063)
	0,50	0,49320 (0,02081)	0,50148 (0,01058)	0,50337 (0,00377)	0,50230 (0,00175)	0,50121 (0,00084)
	0,90	0,81749 (0,01400)	0,85312 (0,00655)	0,88216 (0,00283)	0,89387 (0,00178)	0,89954 (0,00115)
	0,99	0,85415 (0,02308)	0,89294 (0,01186)	0,92819 (0,00491)	0,94668 (0,00249)	0,96037 (0,00121)
Bootstrap	0.01	0.00992 (0.00096)	0.01000 (0.00049)	0.01018 (0.00020)	0.01013 (0.00010)	0.01012 (0.00005)
	0.10	0.09746 (0.00791)	0.09970 (0.00409)	0.10009 (0.00164)	0.09993 (0.00082)	0.10007 (0.00041)
	0.20	0.19446 (0.01290)	0.19837 (0.00665)	0.19976 (0.00262)	0.19977 (0.00131)	0.19994 (0.00065)
	0.50	0.49647 (0.01868)	0.50011 (0.00901)	0.50065 (0.00353)	0.50007 (0.00178)	0.50012 (0.00088)
	0.90	0.89517 (0.01990)	0.90321 (0.01177)	0.90473 (0.00574)	0.90256 (0.00314)	0.90074 (0.00152)
	0.99	0.95119 (0.01453)	0.96522 (0.00742)	0.97635 (0.00317)	0.98100 (0.00170)	0.98475 (0.00090)

```

rmmlk<-function(ssize,rho,K) { # simulate
  R<-runif(ssize,0,1)
  c<-1-rho^(K+1)
  logrho<-log(rho)
  x<-log(1-R*c)%/logrho
  return(x)}
>
> set.seed(13579)
> samp<-rmmlk(ssize=10, rho=0.20, K=5)
> samp
[1] 0 1 0 1 0 0 0 2 1 0

```

LISTING 5: Sample generation for  $M/M/1/K$  queues.

```

MtCaRoMM1K<-function(ssize, rho, K, fEst,...) {
  rep<-10000
  samp<-numeric(ssize)
  est<-numeric(rep)
  for (i in 1:rep) {
    samp<-rmmlk(ssize, rho, K)
    oldseed<-GlobalEnv$.Random.seed
    est [i]<-fEst (K, samp,...)
    GlobalEnv$.Random.seed<-oldseed
  }
  return(c(mean(est), var(est)))}

```

LISTING 6: Monte Carlo simulation.

$$\frac{1 - \rho^X}{c} < R \leq \frac{1 - \rho^{X+1}}{c} \implies \frac{\log(1 - Rc)}{\log(\rho)} - 1 \leq X < \frac{\log(1 - Rc)}{\log(\rho)}. \quad (23)$$

Therefore,

$$X = \left\lceil \frac{\log(1 - Rc)}{\log(\rho)} - 1 \right\rceil, \quad (24)$$

where  $\lceil y \rceil$  is the ceiling function; that is, its value is the least integer that is not inferior to  $y$ .

Listing 5 presents the R implementation [17] used in this study with a sample call.

### 3. Computational Results

To analyze the performance of the estimators, 10,000 Monte Carlo replications were made using the R code [17] shown in Listing 6. Note that  $fEst(K, samp, \dots)$  can be any of the

TABLE 2: Mean estimates for  $\rho$  and the mean squared error (MSE, in parentheses) for  $K = 20$ .

Estimate	$\rho$	$n$				
		10	20	50	100	200
MLE	0.01	0.00928 (0.00084)	0.00960 (0.00045)	0.01000 (0.00019)	0.01003 (0.00010)	0.01007 (0.00005)
	0.10	0.09158 (0.00710)	0.09609 (0.00383)	0.09844 (0.00159)	0.09906 (0.00081)	0.09964 (0.00041)
	0.20	0.18345 (0.01178)	0.19186 (0.00631)	0.19684 (0.00255)	0.19827 (0.00128)	0.19919 (0.00064)
	0.50	0.47371 (0.01424)	0.48785 (0.00666)	0.49543 (0.00256)	0.49750 (0.00127)	0.49885 (0.00062)
	0.90	0.89316 (0.00326)	0.89758 (0.00151)	0.89922 (0.00057)	0.89956 (0.00029)	0.89979 (0.00014)
	0.99	0.97283 (0.00156)	0.97979 (0.00073)	0.98507 (0.00030)	0.98728 (0.00017)	0.98877 (0.00009)
Beta	0.01	0.09009 (0.00692)	0.05374 (0.00226)	0.02864 (0.00052)	0.01953 (0.00018)	0.01487 (0.00007)
	0.10	0.15477 (0.00743)	0.12978 (0.00388)	0.11242 (0.00159)	0.10616 (0.00080)	0.10320 (0.00041)
	0.20	0.22893 (0.00850)	0.21530 (0.00529)	0.20632 (0.00237)	0.20303 (0.00124)	0.20157 (0.00063)
	0.50	0.47818 (0.01132)	0.48902 (0.00599)	0.49557 (0.00245)	0.49753 (0.00125)	0.49885 (0.00062)
	0.90	0.88374 (0.00296)	0.89484 (0.00142)	0.89880 (0.00058)	0.89938 (0.00029)	0.89970 (0.00014)
	0.99	0.94835 (0.00246)	0.96296 (0.00107)	0.97492 (0.00037)	0.98068 (0.00017)	0.98467 (0.00008)
Jeffreys	0,01	0,05049 (0,00219)	0,03182 (0,00084)	0,01931 (0,00026)	0,01476 (0,00012)	0,01244 (0,00005)
	0,10	0,11678 (0,00489)	0,10981 (0,00320)	0,10425 (0,00149)	0,10203 (0,00078)	0,10113 (0,00040)
	0,20	0,19205 (0,00801)	0,19570 (0,00503)	0,19866 (0,00233)	0,19927 (0,00123)	0,19970 (0,00063)
	0,50	0,46191 (0,01540)	0,48175 (0,00769)	0,49434 (0,00302)	0,49828 (0,00152)	0,50063 (0,00074)
	0,90	0,88365 (0,00296)	0,89481 (0,00142)	0,89880 (0,00058)	0,89938 (0,00029)	0,89970 (0,00014)
	0,99	0,94829 (0,00247)	0,96295 (0,00107)	0,97493 (0,00037)	0,98068 (0,00017)	0,98467 (0,00008)
Bootstrap	0.01	0.00993 (0.00096)	0.01000 (0.00049)	0.01018 (0.00020)	0.01013 (0.00010)	0.01012 (0.00005)
	0.10	0.09763 (0.00795)	0.09976 (0.00409)	0.10010 (0.00164)	0.09994 (0.00082)	0.10007 (0.00041)
	0.20	0.19462 (0.01281)	0.19846 (0.00661)	0.19982 (0.00260)	0.19982 (0.00130)	0.19997 (0.00065)
	0.50	0.49385 (0.01404)	0.49896 (0.00659)	0.50019 (0.00254)	0.49995 (0.00127)	0.50008 (0.00062)
	0.90	0.90117 (0.00310)	0.90065 (0.00147)	0.90021 (0.00056)	0.90004 (0.00028)	0.90002 (0.00014)
	0.99	0.98215 (0.00147)	0.98590 (0.00076)	0.98872 (0.00035)	0.98963 (0.00020)	0.99023 (0.00012)

implemented estimation functions for  $\rho$  (MLE, Bayesian, and bootstrap corrected MLE) and that the state of the random seed `GlobalEnv$.Random.seed` was stored immediately before the function `fEst(K, samp, ...)`, which can be stochastic, and was reloaded immediately after its call to ensure that the samples generated for the estimates were the same for all estimators.

**3.1. Simulation Results.** Samples were generated from (24) and Listing 5, for sizes  $n \in \{10; 20; 50; 100; 200\}$  and traffic intensities  $\rho \in \{0.01; 0.10; 0.20; 0.50; 0.90; 0.99\}$ . In these scenarios, averages of 10,000 Monte Carlo replications were computed (via code from Listing 6), and point estimates of  $\rho$  were computed by

- (i) the MLE via numerical maximization of the likelihood function, (2), and code from Listing 1;
- (ii) the Bayesian method via an *a priori* Beta(1.0, 1.0) distribution and an average of 5,000 samples from the *a posteriori* distribution, (5), obtained by the function `arms` [23] from R's HI package [20] via code from Listing 2;
- (iii) the Bayesian method via Jeffreys prior distribution and an average of 1,000 samples from the *a posteriori* distribution, (14), obtained by the function `arms` [23] from R's HI package [20] via code from Listing 3;

- (iv) the bootstrap corrected MLE, (16), and code from Listing 4. Additionally, mean squared errors (MSEs) were calculated.

The results are shown in Tables 1, 2, and 3 and summarized in Figures 3, 4, and 5. In Figures 3, 4, and 5, the average estimation error and the mean squared error (MSE), defined as  $\text{MSE}(\hat{\rho}) = \text{Var}_{\hat{\rho}}(\hat{\rho}) + \text{Bias}^2(\hat{\rho}, \rho)$ , are shown as functions of both the traffic intensity,  $\rho$  (averaged over all sample sizes), and the sample size,  $n$  (averaged over all traffic intensities).

For queues with capacity  $K = 5$  (Figure 3), an approximately constant average estimation error is observed for the MLE and the bootstrap corrected MLE, except when  $\rho \approx 1.0$  (Figure 3(a)). The Bayesian estimators (beta prior and Jeffreys' prior) did not show equivalent performance; their estimates tended to overestimate the true value (positive error) when  $\rho < 0.5$  and to underestimate otherwise. Regarding the sample size,  $n$ , all of the estimators showed a monotonic decrease in error (Figure 3(c)). From the MSE side, both Bayesian estimators presented themselves generally as the best alternative because they presented the lowest values most of the time. Although the bootstrap corrected MLE presented the lowest bias, the method achieves this performance at the cost of high variability, as reflected by its highest MSEs.

When the queue capacity was increased slightly to  $K = 20$  (Figure 4), a very similar behavior was noted. However, the

TABLE 3: Mean estimates for  $\rho$  and the mean squared error (MSE, in parentheses) for  $K = 80$ .

Estimate	$\rho$	$n$				
		10	20	50	100	200
MLE	0.01	0.00928 (0.00084)	0.00960 (0.00045)	0.01000 (0.00019)	0.01003 (0.00010)	0.01007 (0.00005)
	0.10	0.09158 (0.00710)	0.09609 (0.00383)	0.09844 (0.00159)	0.09906 (0.00081)	0.09964 (0.00041)
	0.20	0.18345 (0.01178)	0.19186 (0.00631)	0.19684 (0.00255)	0.19827 (0.00128)	0.19919 (0.00064)
	0.50	0.47366 (0.01422)	0.48783 (0.00666)	0.49543 (0.00255)	0.49750 (0.00127)	0.49885 (0.00062)
	0.90	0.89001 (0.00144)	0.89551 (0.00057)	0.89835 (0.00020)	0.89915 (0.00010)	0.89960 (0.00005)
	0.99	0.98737 (0.00016)	0.98903 (0.00008)	0.98982 (0.00003)	0.98995 (0.00002)	0.98998 (0.00001)
Beta	0.01	0.09009 (0.00692)	0.05374 (0.00226)	0.02864 (0.00052)	0.01953 (0.00018)	0.01487 (0.00007)
	0.10	0.15477 (0.00743)	0.12978 (0.00388)	0.11242 (0.00159)	0.10616 (0.00080)	0.10320 (0.00041)
	0.20	0.22892 (0.00850)	0.21530 (0.00529)	0.20632 (0.00237)	0.20304 (0.00124)	0.20157 (0.00063)
	0.50	0.47779 (0.01125)	0.48892 (0.00597)	0.49556 (0.00245)	0.49753 (0.00125)	0.49885 (0.00062)
	0.90	0.88221 (0.00180)	0.89167 (0.00065)	0.89680 (0.00021)	0.89837 (0.00010)	0.89921 (0.00005)
	0.99	0.98149 (0.00017)	0.98570 (0.00007)	0.98856 (0.00003)	0.98953 (0.00002)	0.98990 (0.00001)
Jeffreys	0,01	0,05225 (0,00231)	0,03334 (0,00089)	0,01998 (0,00027)	0,01506 (0,00011)	0,01253 (0,00005)
	0,10	0,11744 (0,00482)	0,11002 (0,00316)	0,10425 (0,00149)	0,10203 (0,00078)	0,10113 (0,00040)
	0,20	0,19224 (0,00794)	0,19572 (0,00502)	0,19866 (0,00233)	0,19927 (0,00123)	0,19970 (0,00063)
	0,50	0,46147 (0,01532)	0,48167 (0,00767)	0,49430 (0,00302)	0,49826 (0,00151)	0,50064 (0,00074)
	0,90	0,88217 (0,00180)	0,89165 (0,00065)	0,89680 (0,00021)	0,89837 (0,00010)	0,89921 (0,00005)
	0,99	0,98148 (0,00017)	0,98570 (0,00007)	0,98856 (0,00003)	0,98953 (0,00002)	0,98990 (0,00001)
Bootstrap	0.01	0.00993 (0.00096)	0.01000 (0.00049)	0.01018 (0.00020)	0.01013 (0.00010)	0.01012 (0.00005)
	0.10	0.09763 (0.00795)	0.09976 (0.00409)	0.10010 (0.00164)	0.09994 (0.00082)	0.10007 (0.00041)
	0.20	0.19462 (0.01281)	0.19846 (0.00661)	0.19982 (0.00260)	0.19982 (0.00130)	0.19997 (0.00065)
	0.50	0.49390 (0.01405)	0.49897 (0.00659)	0.50019 (0.00254)	0.49995 (0.00127)	0.50008 (0.00062)
	0.90	0.89910 (0.00118)	0.89987 (0.00051)	0.90008 (0.00019)	0.90002 (0.00009)	0.90003 (0.00005)
	0.99	0.99000 (0.00016)	0.99042 (0.00009)	0.99031 (0.00004)	0.99012 (0.00002)	0.99002 (0.00001)

difficulty of estimation for traffic intensities  $\rho \approx 1.0$  seemed to decrease, and the highest MSEs occurred when  $\rho \approx 0.5$ . The Bayesian estimators maintained the performance presented earlier, for  $K = 5$ ; that is, the estimates have positive bias for  $\rho < 0.5$  and negative bias otherwise. The errors of all of the estimators converged to zero as the sample size grew. The bootstrap corrected MLE presented an average estimation error near zero for samples  $n \geq 50$ . However, from the point of view of the MSE, the smaller values were again obtained using the Bayesian methods (beta and Jeffreys' prior).

Finally, for queues with  $K = 80$  (Figure 5), the observed behavior could be considered, for practical purposes, as being equal to that of an infinite Markovian queue in terms of the average estimation error and the MSE. This behavior was observed for infinite Markovian queues ( $M/M/1$  queues) [15]. This finding is merely evidence of the correctness of our implementations and the quality of the computational results presented.

Additionally, computational experiments were performed for the estimators for  $L$ , (3), and for  $L_q$ , (4), and for their bootstrap corrected versions, with 1,000 resamplings, and  $K = 20$ . Table 4 and Figure 6 show the results obtained for  $L \in \{0.5, 1, 2, 4, 8, 16\}$ , for sample sizes  $n \in \{10, 20, 50, 100\}$  and averages of 10,000 Monte Carlo replications. Similarly for  $L_q$ , the results are presented in Table 5 and Figure 7. In summary, at an extra cost of the bootstrap method and without inflation of the MSEs, the

researcher may achieve with samples of size  $n = 10$  estimates for  $L$  and  $L_q$  with the same average error of the MLE for samples of size  $n = 100$ , a reduction that is relevant in practical terms because it may lead to reduction in time and cost to obtain the estimates. Note that the bootstrap method always provides smaller errors and MSEs than the MLE method for all estimates of  $L$  and  $L_q$ , even when  $\rho > 1$ . Also note the jump up and down in the errors and MSEs when  $\rho$  transitions from  $\rho < 1$  to  $\rho > 1$ .

Finally, to illustrate the ease of use of the bootstrap in the interval estimation of the traffic intensity  $\rho$ , computational experiments were performed. The length and coverage of empirical bootstrap intervals computed from (18) and from a normal distribution approximation (*i.e.*,  $\widehat{\Theta} \pm z_{\alpha/2} \sigma_{\widehat{\Theta}}$ , where  $z_{\gamma}$  is the  $\gamma$ 100%th percentile of the standard normal distribution and the standard deviation  $\sigma_{\widehat{\Theta}}$  was estimated also by bootstrapping) were evaluated for  $\rho \in \{0.10, 0.20, 0.50, 0.90\}$ , for sample sizes  $n \in \{10, 20, 50, 100, 200\}$ , averages of 10,000 Monte Carlo replications, and  $K = 20$ . The satisfactory performance of the bootstrap was demonstrated, as presented in Table 6, with the coverages approaching the nominal confidence of 95% (that is,  $1 - \alpha = 0.95$ ) as the sample sizes increase.

**3.2. Numerical Example.** To better illustrate an application of the method, a numerical application based on the data considered in Table 7, collected in a large supermarket network

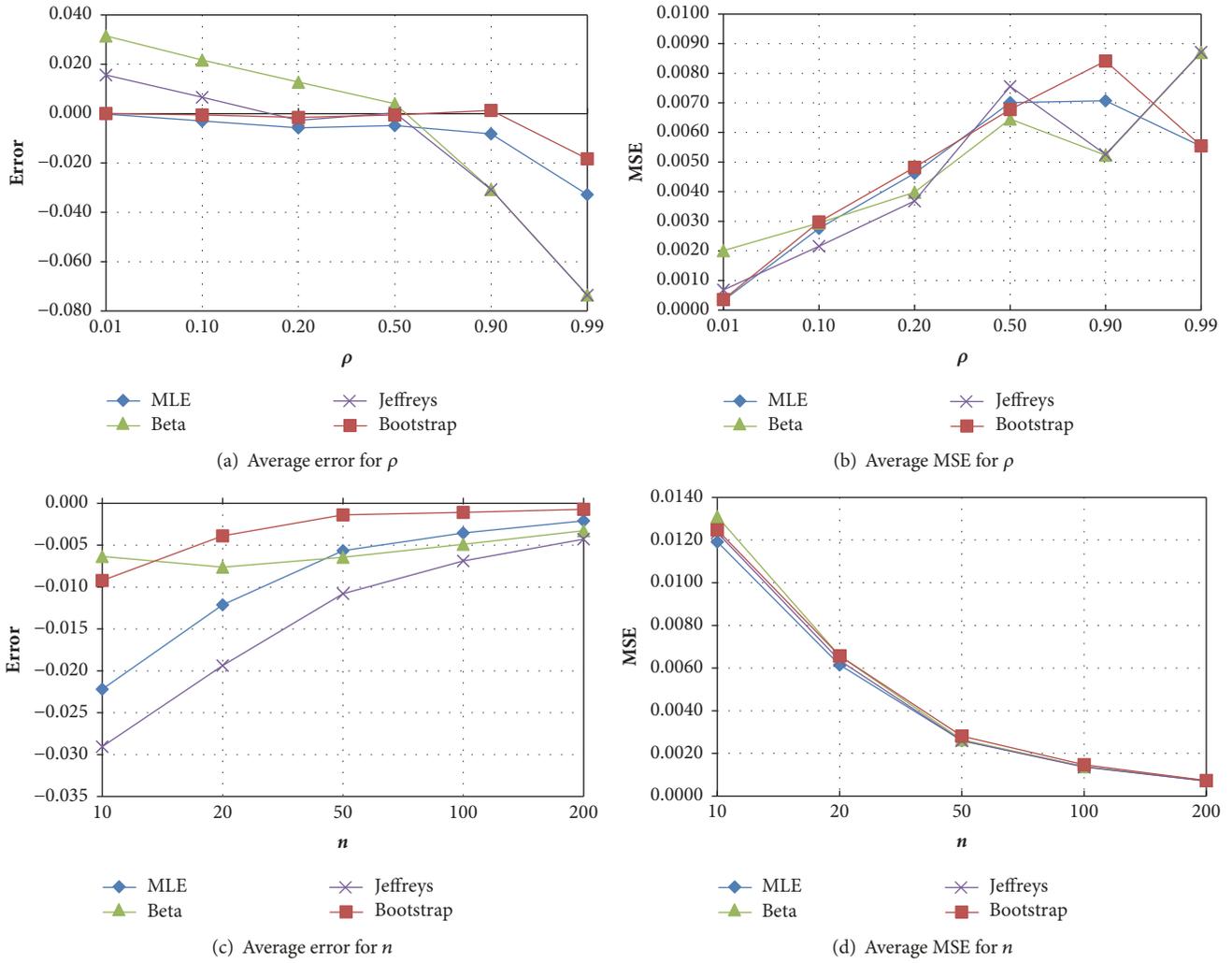


FIGURE 3: Performance of estimators for  $\rho$  and  $K = 5$ .

TABLE 4: Mean estimates for  $L$  and the mean squared error (MSE, in parentheses) for  $K = 20$ .

Estimator	$L$	$\rho$	$n$			
			10	20	50	100
MLE	0.50	0.33333	0.45010 (0.01698)	0.47586 (0.00806)	0.49098 (0.00303)	0.49523 (0.00150)
	1.00	0.50000	0.90008 (0.02353)	0.95255 (0.00877)	0.98190 (0.00286)	0.99007 (0.00137)
	2.00	0.66714	1.80441 (0.04755)	1.90696 (0.01287)	1.96438 (0.00283)	1.98120 (0.00112)
	4.00	0.80959	3.72455 (0.08082)	3.87956 (0.01669)	3.95673 (0.00268)	3.97720 (0.00092)
	8.00	0.94574	7.88487 (0.01625)	7.97002 (0.00230)	7.99390 (0.00057)	7.99583 (0.00029)
	16.0	1.23520	16.2965 (0.09872)	16.1779 (0.03702)	16.0743 (0.00747)	16.0377 (0.00238)
Bootstrap	0.50	0.33333	0.48573 (0.01586)	0.49686 (0.00777)	0.50033 (0.00299)	0.50009 (0.00149)
	1.00	0.50000	0.97570 (0.01460)	0.99583 (0.00661)	1.00076 (0.00254)	0.99982 (0.00127)
	2.00	0.66714	1.96026 (0.01062)	1.99245 (0.00416)	2.00066 (0.00154)	1.99975 (0.00076)
	4.00	0.80959	3.97317 (0.00515)	4.00156 (0.00204)	4.00475 (0.00081)	4.00113 (0.00039)
	8.00	0.94574	7.96978 (0.00350)	8.00755 (0.00138)	8.00792 (0.00059)	8.00277 (0.00027)
	16.0	1.23520	16.1315 (0.03011)	16.0376 (0.00699)	16.0043 (0.00188)	16.0037 (0.00095)

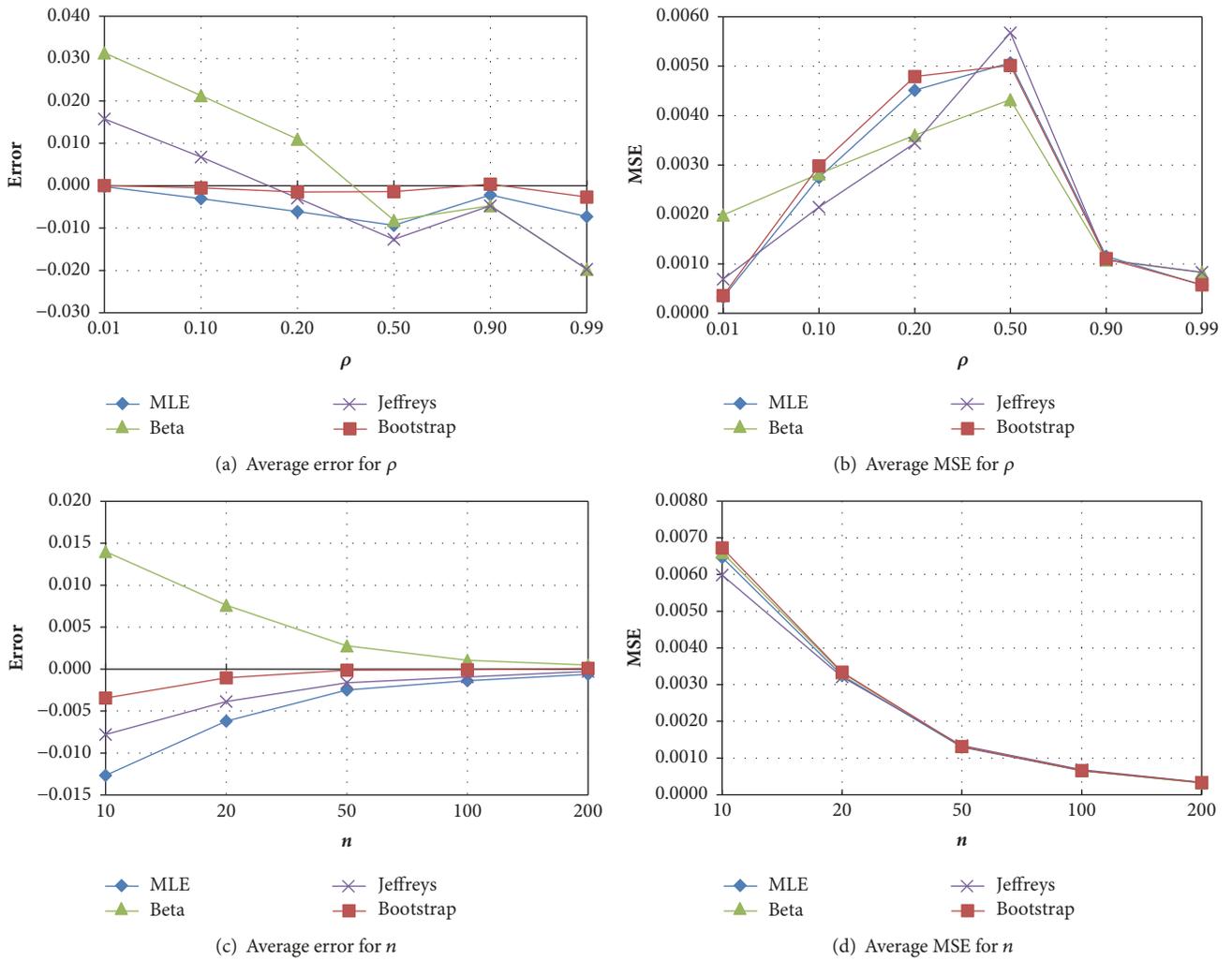


FIGURE 4: Performance of estimators for  $\rho$  and  $K = 20$ .

TABLE 5: Mean estimates for  $L_q$  and the mean squared error (MSE, in parentheses) for  $K = 20$ .

Estimator	$L_q$	$\rho$	$n$			
			10	20	50	100
MLE	0.50	0.50000	0.42637 (0.01897)	0.46470 (0.00776)	0.48647 (0.00272)	0.49256 (0.00132)
	1.00	0.61818	0.86547 (0.02882)	0.93515 (0.00918)	0.97534 (0.00248)	0.98726 (0.00108)
	2.00	0.73449	1.77618 (0.05718)	1.89501 (0.01417)	1.96115 (0.00266)	1.97942 (0.00099)
	4.00	0.84574	3.74340 (0.06999)	3.89252 (0.01340)	3.96186 (0.00214)	3.97930 (0.00077)
	8.00	0.97126	7.94718 (0.00577)	7.99972 (0.00141)	8.00478 (0.00056)	8.00250 (0.00028)
	16.0	1.32722	16.1536 (0.03518)	16.1286 (0.02392)	16.0683 (0.00800)	16.0254 (0.00210)
Bootstrap	0.50	0.50000	0.48185 (0.01434)	0.49688 (0.00660)	0.50057 (0.00254)	0.49986 (0.00127)
	1.00	0.61818	0.97175 (0.01146)	0.99492 (0.00493)	1.00110 (0.00185)	1.00049 (0.00092)
	2.00	0.73449	1.96521 (0.00787)	1.99479 (0.00303)	2.00236 (0.00113)	2.00027 (0.00056)
	4.00	0.84574	3.97906 (0.00409)	4.00426 (0.00174)	4.00494 (0.00069)	4.00067 (0.00033)
	8.00	0.97126	7.96303 (0.00394)	8.00653 (0.00136)	8.00732 (0.00058)	8.00362 (0.00028)
	16.0	1.32722	16.1202 (0.03015)	16.0862 (0.01687)	16.0226 (0.00418)	16.0228 (0.00229)

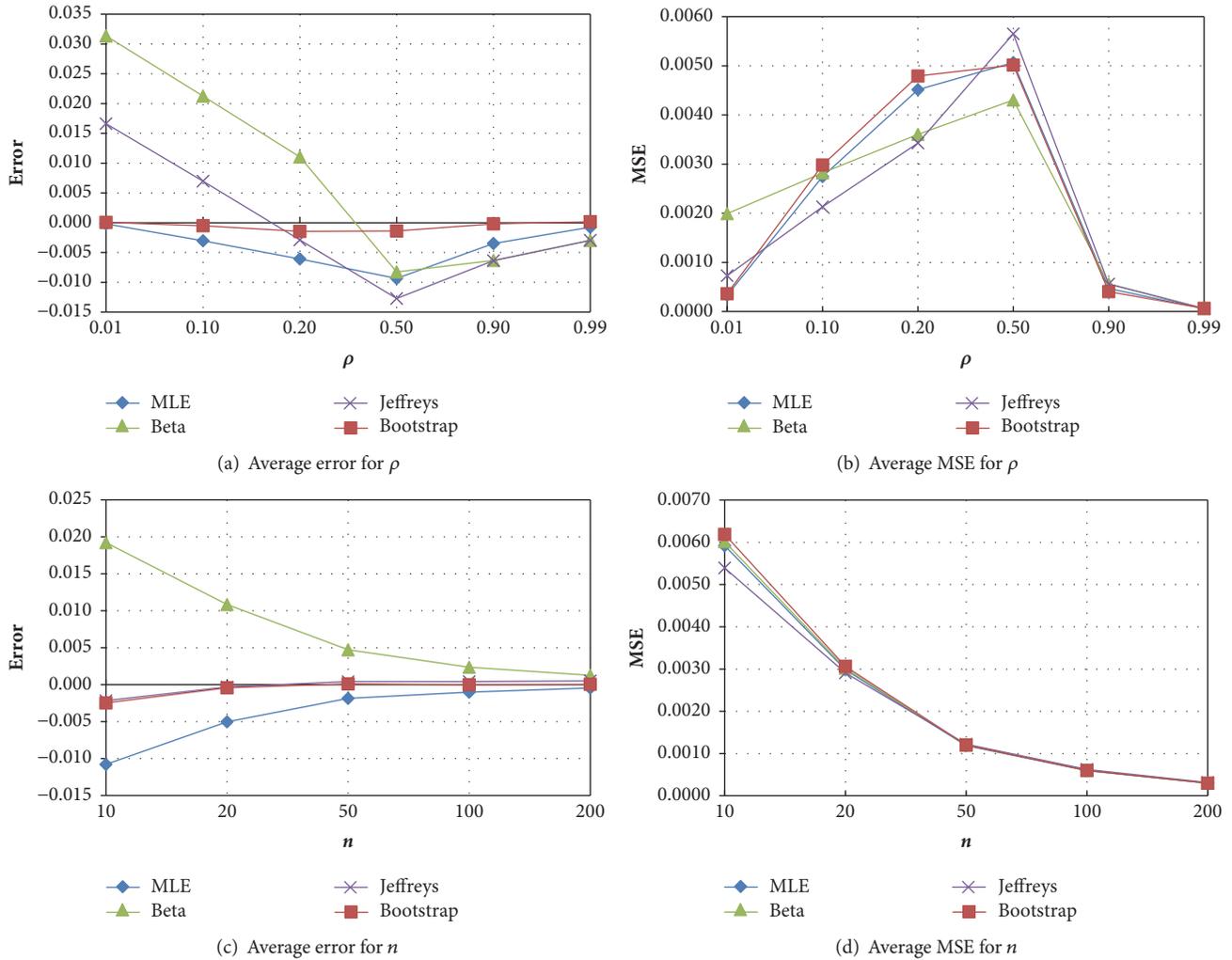


FIGURE 5: Performance of estimators for  $\rho$  and  $K = 80$ .

TABLE 6: Average length ( $\mathcal{L}$ ) and coverage ( $\mathcal{C}$ ) of 95% confidence intervals for  $\rho$  and  $K = 20$ .

Method	$\rho$	$n = 10$		$n = 20$		$n = 50$		$n = 100$		$n = 200$	
		$\mathcal{L}$	$\mathcal{C}$								
Empirical Bootstrap	0.10	0.189	0.626	0.185	0.612	0.143	0.885	0.106	0.906	0.077	0.933
	0.20	0.308	0.619	0.265	0.797	0.186	0.902	0.135	0.923	0.097	0.940
	0.50	0.402	0.836	0.294	0.892	0.191	0.924	0.137	0.938	0.097	0.943
	0.90	0.207	0.835	0.148	0.911	0.094	0.958	0.066	0.953	0.046	0.952
Normal Approximation	0.10	0.219	0.636	0.200	0.865	0.146	0.890	0.107	0.911	0.077	0.936
	0.20	0.335	0.868	0.274	0.835	0.188	0.909	0.136	0.934	0.098	0.942
	0.50	0.410	0.885	0.298	0.916	0.193	0.934	0.137	0.942	0.098	0.947
	0.90	0.213	0.906	0.149	0.937	0.094	0.953	0.066	0.950	0.046	0.950

in a region of interest [12], is provided. The goal is to evaluate traffic intensity, which, for managerial reasons, should not exceed 87%; if it does, users may leave. The data comprise 200 random observations of the number of customers in the system at random times sufficiently spaced and previously defined by the person responsible for collecting the data to

avoid correlation. The observed values (V) and frequency (F) are presented in Table 7. For instance, of the 200 observations, at 8 times, no customers were found in the system; at 21 times, only 1 customer was found, and so on.

The estimates are shown in Table 8; they were calculated using the MLE (Listing 1) and the Bayesian estimators

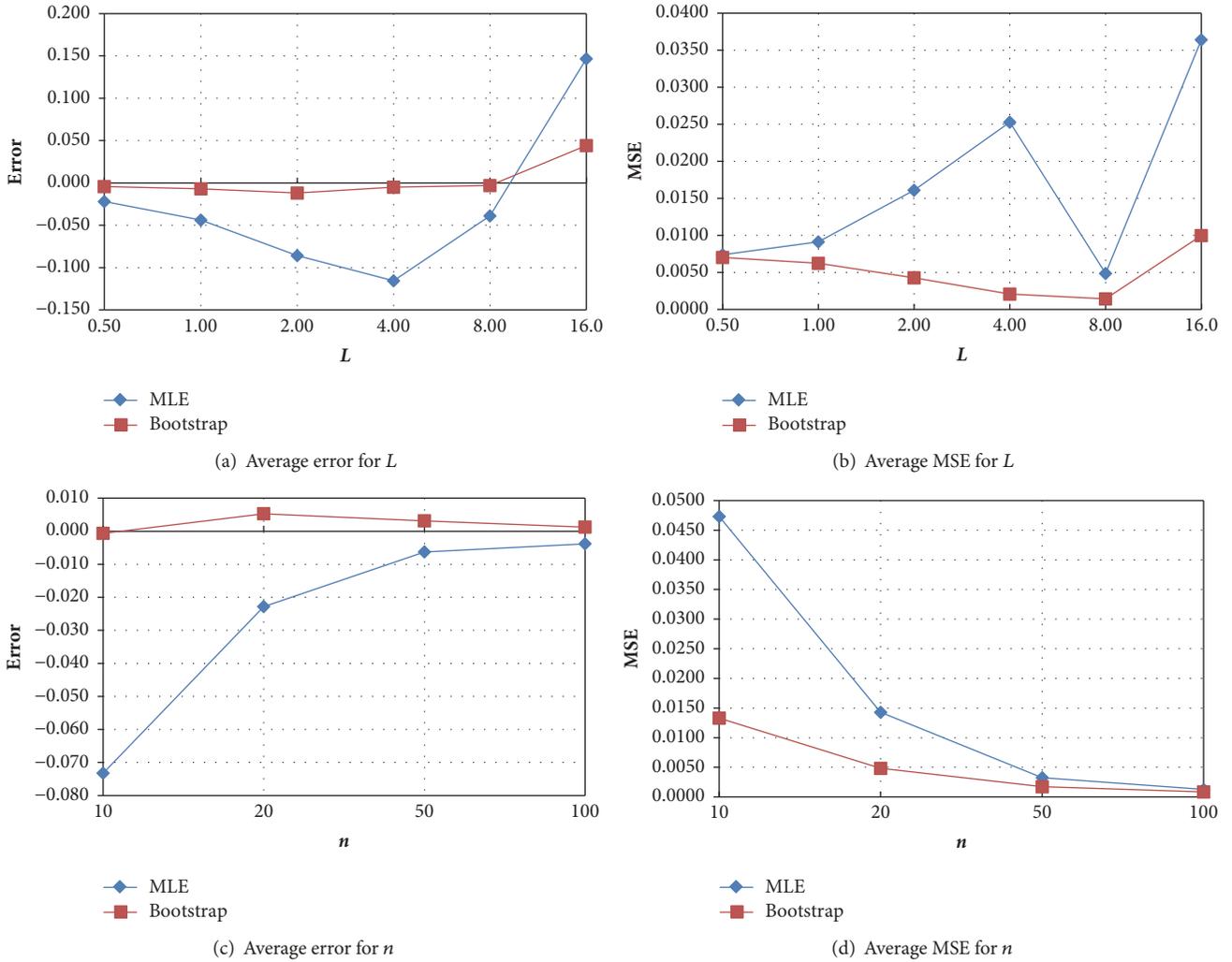


FIGURE 6: Performance of estimates for  $L$ .

TABLE 7: Observed values ( $V$ ) and frequency ( $F$ ) for a sample of size  $n = 200$  with  $K = 14$  [12].

$V$	$F$
0	8
1	21
2	27
3	39
4	29
5	28
6	16
7	15
8	6
9	5
10	2
11	1
14	3

(Listings 2 and 3) based on an *a priori* Beta(1.0, 1.0) and 5,000 samples and Jeffreys’ distribution and 1,000 samples from

TABLE 8: Point estimates for the numerical example.

Estimate	MLE	Bayes	Jeffreys	Bootstrap Corrected
$\hat{\rho}$	0.8405396	0.8403625	0.8403625	0.8405683

the *a posteriori* distribution; the bootstrap corrected MLE (Listing 4) was based on 1,000 resamplings. The complete script is shown in Listing 7.

According to the results presented in the previous section, the Bayesian estimates should be the most reliable. It is notable that the system utilization seemed to be below the target ( $\hat{\rho} = 0.8405396 < 0.87$ ). Note that the analysis is based only on counts of the number of users in the system. It is not necessary to estimate the arrival and service rates separately to determine  $\rho$ .

#### 4. Conclusions and Final Observations

The problem of traffic intensity estimation in finite Markov queues ( $M/M/1/K$  queues) is presented as quite challenging. In fact, no estimator is absolutely superior to another in all

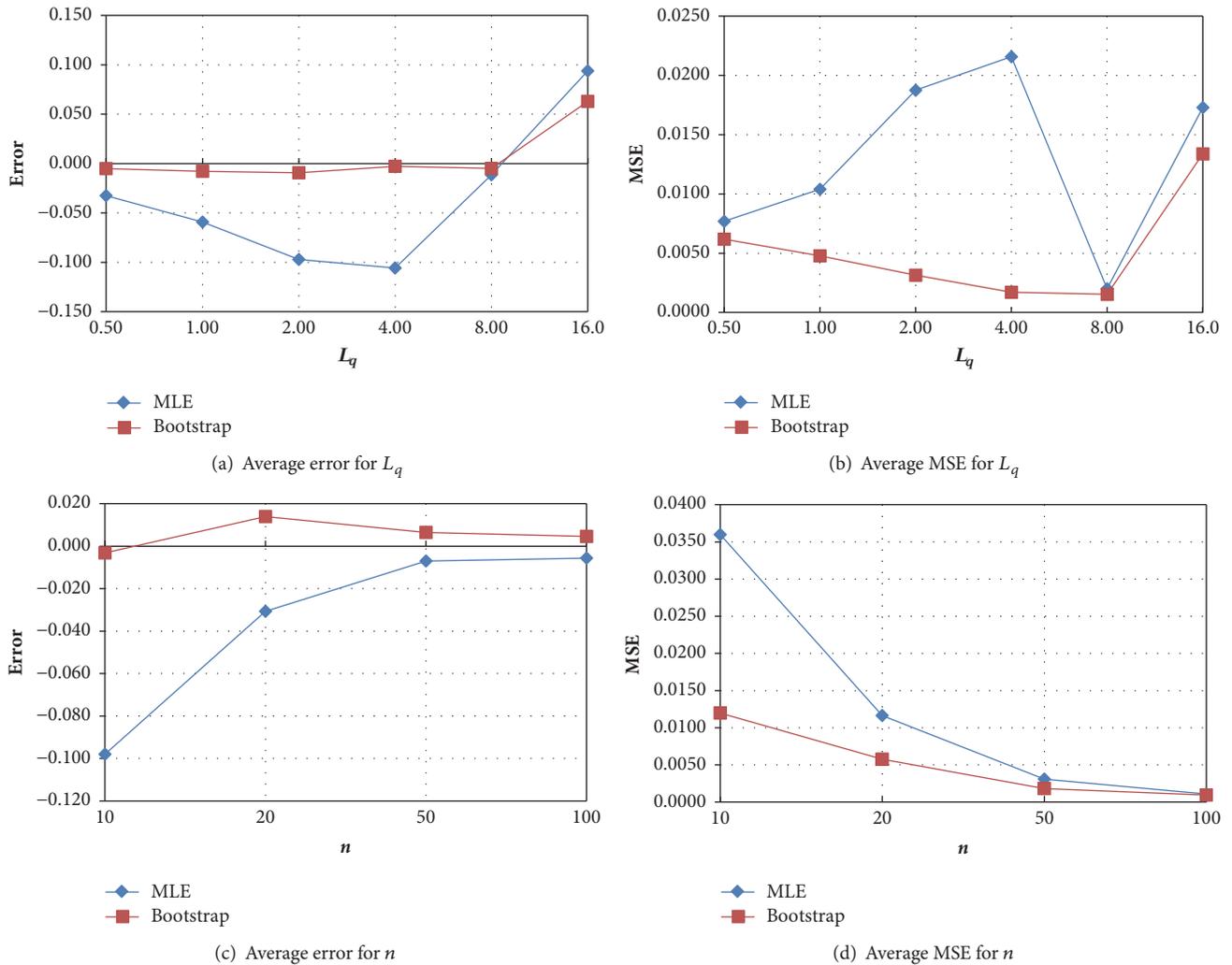


FIGURE 7: Performance of estimates for  $L_q$ .

```

# read sample
samp<-c(rep(0, 8), rep(1, 21), rep(2, 27), rep(3, 39), rep(4, 29), rep(5, 28), rep(6, 16),
rep(7, 15), rep(8, 6), rep(9, 5), rep(10, 2), rep(11, 1), rep(14, 3))
# MLE estimate
K<-14
hatrhoMLE<-MLERoMM1K(K, samp)
# Bayesian estimate
a<-1.0
b<-1.0
set.seed(13579)
hatrhoBayes<-EBaRoMM1K(K, samp, a, b)
# Bayesian Jeffreys estimate
set.seed(13579)
hatrhoJe<-EJeRoMM1K(K, samp)
# Bootstrap corrected estimate
set.seed(13579)
hatrhoBoot<-EBoRoMM1K(K, samp)
c(hatrhoMLE, hatrhoBayes, hatrhoJe, hatrhoBoot)
> [1] 0.8405396 0.8403625 0.8403625 0.8405683
    
```

LISTING 7: Estimations from real data.

parametric space. Although the estimates of the MLE and the bootstrap corrected MLE exhibit less bias, the Bayesian estimates (beta and Jeffreys' prior) present the lowest MSE in general. Perhaps due to the skewness of the *a posterior* distribution, the Bayesian estimators do not present low bias. In general, for sample size  $n = 50$  and queues with  $K \geq 20$ , the average estimation error is less than 0.005; this value was only exceeded by the Bayesian estimator for queues with total capacity  $K = 5$ .

In regard to the behavior of the average estimation error and the average MSE as functions of the traffic intensity  $\rho$ , the major errors are observed when the sample size is small ( $n \leq 20$ ) and the traffic intensities are  $\rho \approx 1.0$ , unlike in the case of the  $M/M/1$  queues, which exhibit higher biases when  $\rho \approx 0.5$ . Perhaps due to the truncation of the number of users to the maximum queue length,  $K$ , systems with high traffic intensities require more computational effort and are the most difficult to estimate.

Finally, it is important to note that, for queues with capacity  $K = 80$ , the system's behavior is similar to that of an infinite Markovian queue (an  $M/M/1$  queue), as expected. That is, the average estimation error is greater, and the MSE is highest when  $\rho \approx 0.5$ .

Future work in this area includes testing other Bayesian point estimators (e.g., the median, because of the asymmetry of the *a posterior* distribution), developing interval estimators, hypothesis testing methods, or even Kernel-based methods [24].

## Data Availability

The data used to support the findings of this study are included within the article.

## Disclosure

The Brazilian government funding agencies mentioned had no role in the study.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

## Acknowledgments

Special thanks are due to Gabriel and Carolina for helping with the algebra. This work was supported by the Brazilian agencies CNPq (*Conselho Nacional de Desenvolvimento Científico e Tecnológico* of the Ministry for Science and Technology) [Grant nos. 304671/2014-2, 305841/2016-5] and FAPEMIG (Fundação de Amparo à Pesquisa do Estado de Minas Gerais) [Grant nos. CEX-PPM-00564-17, APQ-02119-15, and BIP-00106-16].

## References

- [1] D. Ma, D. Wang, Y. Bie, F. Sun, and S. Jin, "A method for queue length estimation in an urban street network based on roll time occupancy data," *Mathematical Problems in Engineering*, vol. 2012, p. 12, 2012, <http://dx.doi.org/10.1155/2012/892575>.
- [2] S. Zhao, S. Liang, H. Liu, and M. Ma, "CTM based real-time queue length estimation at signalized intersection," *Mathematical Problems in Engineering*, vol. 2015, Article ID 328712, 12 pages, 2015.
- [3] C. Armero and M. J. Bayarri, "Bayesian prediction in  $M/M/1$  queues," *Queueing Systems*, vol. 15, no. 1-4, pp. 401-417, 1994.
- [4] C. Armero and M. J. Bayarri, "Queues," in *Proceedings of the International Encyclopedia of the Social Behavioral Sciences*, J. D. Wright, Ed., pp. 784-789, Oxford, UK, 2015.
- [5] C. Armero and D. Conesa, "Inference and prediction in bulk arrival queues and queues with service in stages," *Applied Stochastic Models in Business and Industry*, vol. 14, no. 1, pp. 35-46, 1998.
- [6] C. Armero and D. Conesa, "Prediction in Markovian bulk arrival queues," *Queueing Systems*, vol. 34, no. 1-4, pp. 327-350, 2000.
- [7] C. Armero and D. Conesa, "Statistical performance of a multiclass bulk production queueing system," *European Journal of Operational Research*, vol. 158, no. 3, pp. 649-661, 2004.
- [8] C. Armero and D. Conesa, "Bayesian hierarchical models in manufacturing bulk service queues," *Journal of Statistical Planning and Inference*, vol. 136, no. 2, pp. 335-354, 2006.
- [9] A. Choudhury and A. C. Borthakur, "Bayesian inference and prediction in the single server Markovian queue," *Metrika. International Journal for Theoretical and Applied Statistics*, vol. 67, no. 3, pp. 371-383, 2008.
- [10] S. Chowdhury and S. P. Mukherjee, "Estimation of traffic intensity based on queue length in a single  $M/M/1$  queue," *Communications in Statistics—Theory and Methods*, vol. 42, no. 13, pp. 2376-2390, 2013.
- [11] S. Chowdhury and S. P. Mukherjee, "Bayes estimation in  $M/M/1$  queues with bivariate prior," *Journal of Statistics and Management Systems*, vol. 19, no. 5, pp. 681-699, 2016.
- [12] F. R. Cruz, R. C. Quinino, and L. L. Ho, "Bayesian estimation of traffic intensity based on queue length in a multi-server  $M/M/S$  queue," *Communications in Statistics—Simulation and Computation*, vol. 46, no. 9, pp. 7319-7331, 2017.
- [13] R. C. Quinino and F. R. B. Cruz, "Bayesian sample sizes in an  $M/M/1$  queueing systems," *The International Journal of Advanced Manufacturing Technology*, vol. 88, no. 1-4, pp. 995-1002, 2017.
- [14] D. Gross, J. F. Shortle, and J. M. Thompson, *Fundamentals of Queueing Theory*, Wiley-Interscience, New York, NY, USA, 4th edition, 2009.
- [15] M. A. C. Almeida and F. R. B. Cruz, "A note on Bayesian estimation of traffic intensity in single-server Markovian queues," *Communications in Statistics - Simulation Computation*, <http://dx.doi.org/10.1080/03610918.2017.1353614>.
- [16] R. L. Thomasson and C. H. Kapadia, "On estimating the parameter of a truncated geometric distribution," *Annals of the Institute of Statistical Mathematics*, vol. 20, pp. 519-523, 1968.
- [17] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2017, <http://www.R-project.org/>.
- [18] N. Mukhopadhyay, *Probability and Statistical Inference*, Marcel Dekker, New York, NY, USA, 2000.
- [19] G. S. Lingappaiah, "Bayes inference in right truncated geometric distribution," *Malaysian Mathematical Society. Bulletin. Second Series*, vol. 15, no. 2, pp. 61-67, 1992.

- [20] G. Petris and L. Tardella, "HI: Simulation from distributions supported by nested hyperplanes, 2013. (original C code for ARMS by Wally R. Gilks; R package version 0.4)," <http://CRAN.R-project.org/package=HI>.
- [21] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*, Monographs on Statistics and Applied Probability, Chapman and Hall, New York, NY, USA, 1993.
- [22] B. Efron and R. Tibshirani, "Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy," *Statistical Science*, vol. 1, no. 1, pp. 54–75, 1986.
- [23] W. R. Gilks, N. G. Best, and K. K. C. Tan, "Adaptive rejection metropolis sampling within gibbs sampling," *Journal of the Royal Statistical Society*, vol. 44, Series C, no. 4, pp. 455–472, 1995.
- [24] G. M. Gontijo, G. S. Atuncar, F. R. B. Cruz, and L. Kerbache, "Performance evaluation and dimensioning of  $GI^X/M/c/N$  systems through kernel estimation," *Mathematical Problems in Engineering*, vol. 2011, Article ID 348262, 20 pages, 2011, <http://dx.doi.org/10.1155/2011/348262>.

